ENZO

An adaptive mesh refinement code for cosmology simulations

Michael Norman (mnorman@cosmos.ucsd.edu)

James Bordner (jbordner@cosmos.ucsd.edu)

Laboratory for Computational Astrophysics

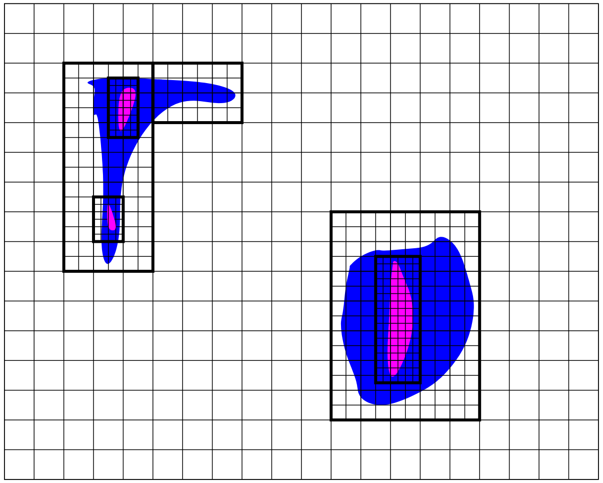
University of California, San Diego

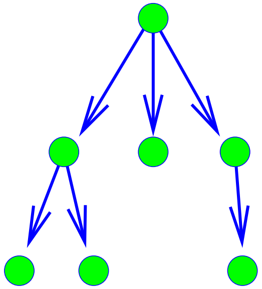
UC San Diego

Outline

- Structured Adaptive Mesh Refinement (SAMR)
- Enzo, a parallel SAMR cosmology code
- Dynamic load balancing
- Performance on a sample run
- Outstanding performance issues

Structured Adaptive Mesh Refinement (SAMR)

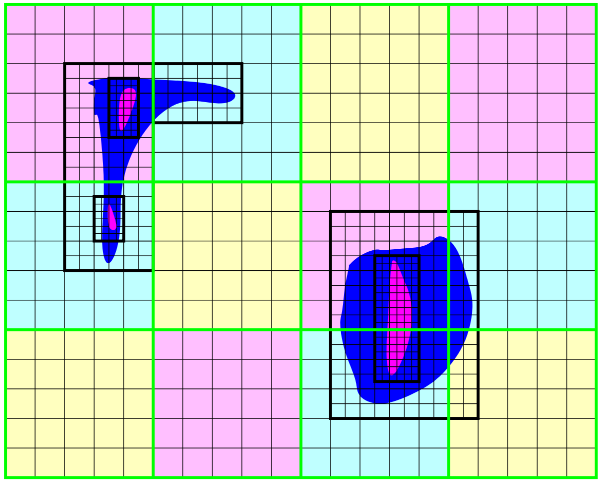




- Berger & Colella, 1989
- For time-dependent shock hydrodynamics with large spacial dynamic range of solution features
- Smaller time steps on finer grids
- These grids are dynamic
- Before advancing a timestep, a grid gets boundary data from its parent
- Afterwards, its parent updates its overlapped region with the fine grid solution

UC San Diego

Enzo, a parallel SAMR cosmology code



- G. Bryan, M. Norman
- gravity + dark matter dynamics + gas dynamics + ionization kinetics + radiative transfer + . . .
- Root grid is distributed among processors (MPI)
- Each refined grid lies in a single processor
- Each processor has global hierarchy information
- Meshes within a level are parallel
- Communication between adjacent levels
- Dynamic load balancing is crucial!

UC San Diego

Dynamic load balancing

- Z. Lan, V. Taylor, G. Bryan
- based on moving-grid and splitting-grid phases
- each processor knows about grids on other processors
- grids moved from over-loaded processors to underloaded processors
- see Z. Lan et al for details

Performance on a sample run

First star formation (G. Bryan, T. Abel, M. Norman)

- over 8000 subgrids at 34 levels of refinement
- 128-bit precision required in parts
- 13 Gflop/s on 64 processors of Blue Horizon
- up to 20GB memory, 50-100 GB disk, “much more mass storage”

component	cost
hydrodynamics	36 %
Poisson solver	17 %
chemistry & cooling	11 %
N-body	1 %
hierarchy rebuild	9 %
boundary conditions	15 %
other overhead	11 %

Outstanding performance issues

Functionality issues:

- MPI implementation limits: hit max # outstanding messages

Performance issues:

- Capability: solve bigger problems
- Turnaround: solve problems faster
- Scaling to 1000s of processors?
- Performance analysis and modeling with dynamic data structures
- Load balancing crucial—how does current scheme scale?

Random issues:

- MPI-2 capabilities look desirable
- MPI + threads may be useful